Evaluation of Continuous Activity Scheduling Planning Execution and Replanning (CASPER).

Dan Berry / Code 582

Evaluation Objectives

- CASPER
  - What is it?
  - How does it work?
  - Applicability & Advantages/disadvantages?

CASPER – What is it?

- **Formal definition :**
  - **CASPER uses iterative repair to support continuous modification and updating of a current working plan in light of changing operating context[1].**

- **CASPER is a on-board scheduler/planner that can accept external (on-board generated) inputs to update the schedule in response to events of opportunity.**

- **Notes :**
  - **ASPEN (Automated Scheduling and Planning Environment) is a ground based planner/scheduler.**
  - **CASPER (*Continuous Activity Scheduling Planning Execution and Replanning*) is a flight based planner/scheduler with the same core as ASPEN.**

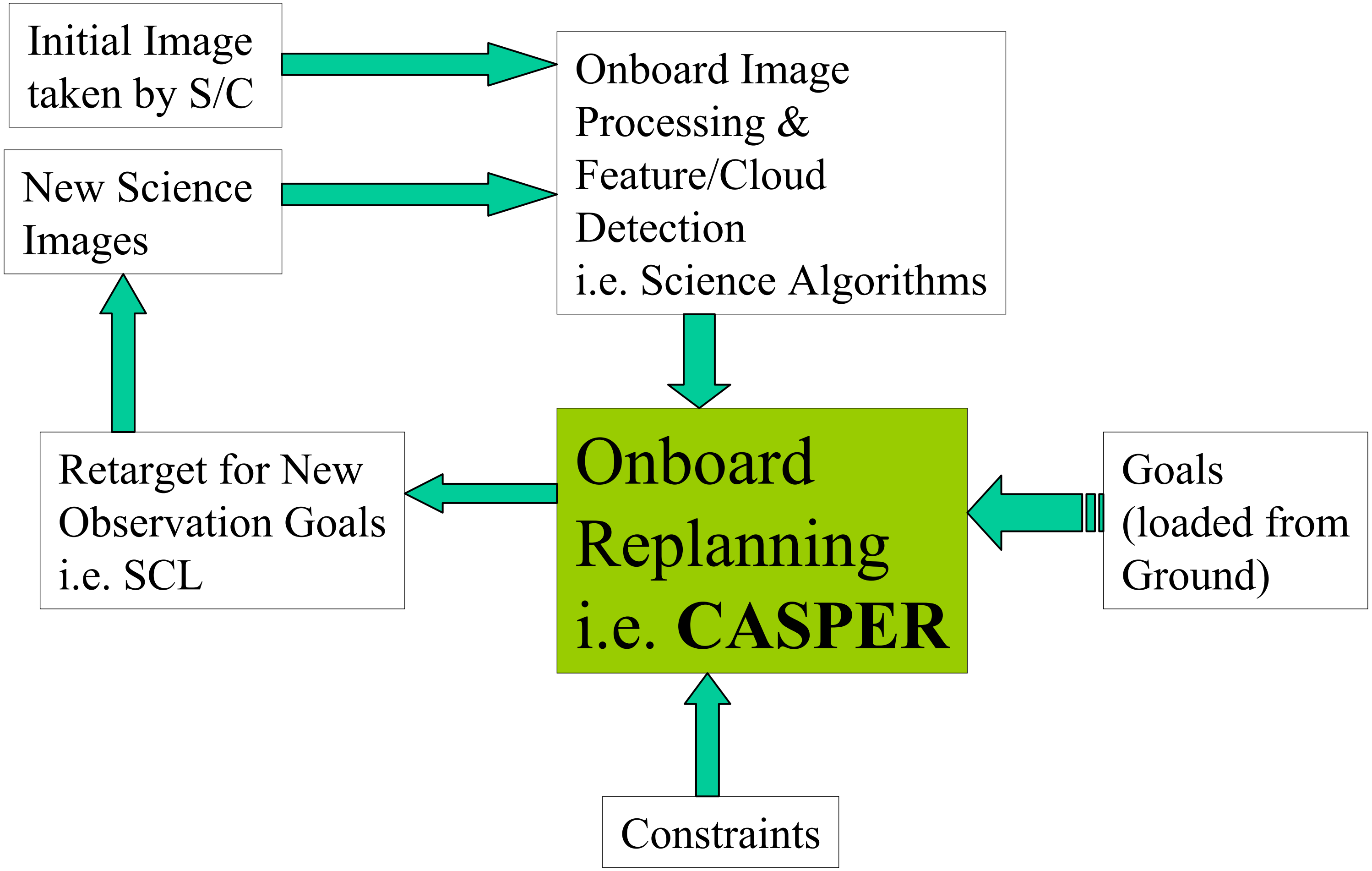
CASPER – Tidbits

- **Developed at JPL.**
  - Planning and Scheduling Artificial Intelligence Group.
  - Available to support future deployments.
    - Previous mission costs $300K - $3M
      - Probably closer to the higher end for full up S/C.
  - Other missions planned :
    - Three Corner Sat (3CS) planned for 2003 (was shuttle mission)
    - TechSat-21 Constellation planned for 2004
- **Written in C/C++.**
  - Acquiring source code is subject to ITAR restrictions.
  - Other documentation (requirements, etc) available.
- **Development Environment – Linux, Solaris, & Windows.**
- **Interfaces with middleware to translate higher level commands into low level S/C commands.**
  - EO-1 experiment uses Spacecraft Command Language (SCL).
- **Nothing special done to testbed in support of CASPER.**
  - EO-1 experiment required updates to GSFC SWB, cmd/tlm database, etc.

ASE – EO-1

- **EO-1**
  - **Launch 11/21/2000**
  - **Currently in extended mission (primary mission objectives have been achieved).**
- **Part of the Autonomous Sciencecraft Experiment (ASE)[2].**
  - **Onboard science algorithms - that will analyze the image data to detect trigger conditions such as science events, interesting features, changes relative to previous observations, and cloud detection for onboard image editing.**
  - **Robust execution management software - using the Spacecraft Command Language (SCL) package to enable event-driven processing and low-level autonomy.**
  - **CASPER software - that will replan activities, including downlink, based on science observations in the previous orbit cycles.**
- **CASPER and SCL are stand alone products. The science algorithms are also a stand alone product but are coupled with CASPER and SCL (scheduled by CASPER and executed in batch mode by SCL).**
  - **CASPER is an interesting product.**
  - **ASE is a really interesting solution.**
    - **Via the ability to autonomously update the schedule.**

ASE – EO-1 Environment



Initial Image taken by S/C

New Science Images

Onboard Image Processing & Feature/Cloud Detection i.e. Science Algorithms

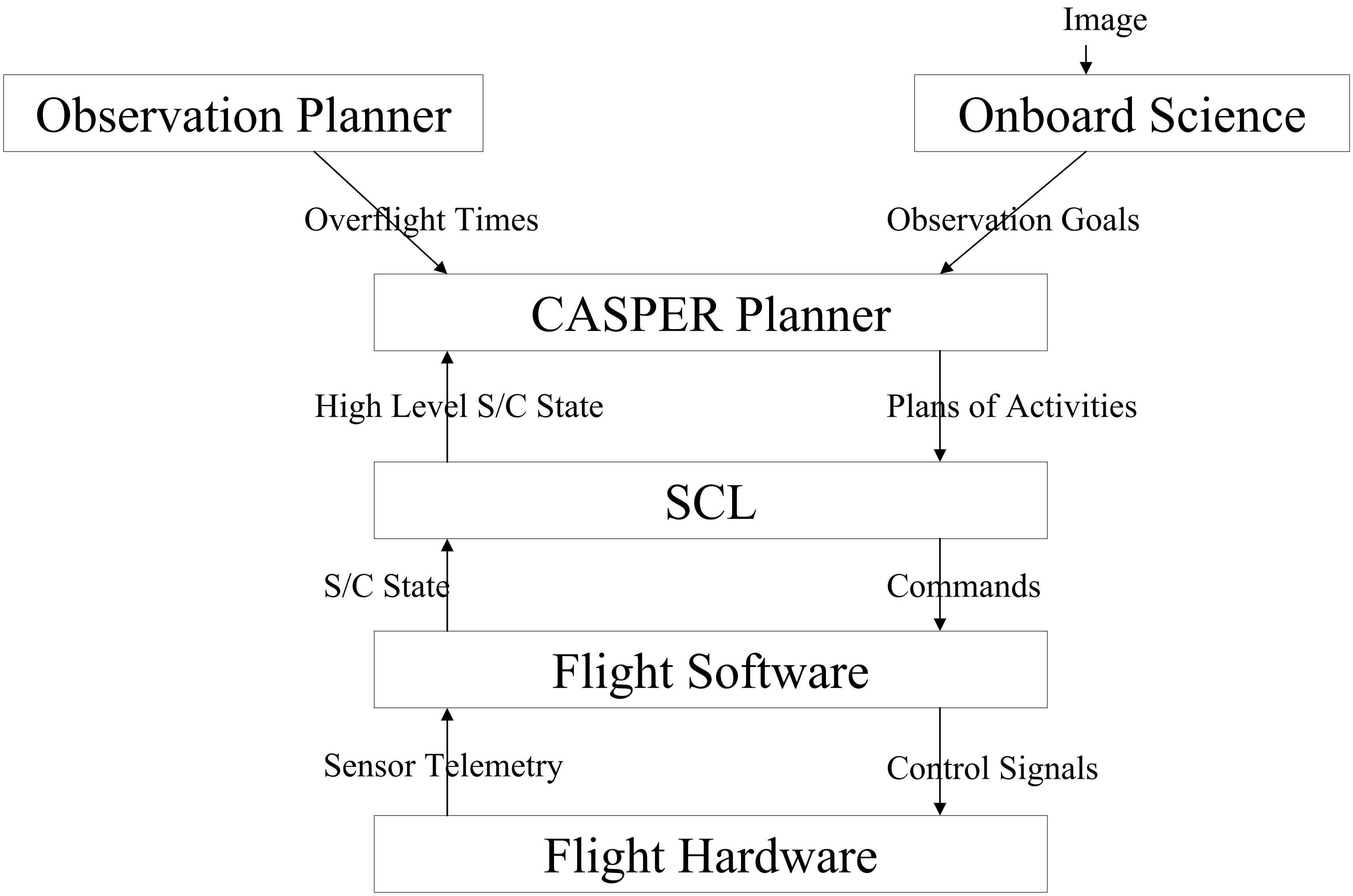
Retarget for New Observation Goals i.e. SCL

Onboard Replanning i.e. **CASPER**

Goals (loaded from Ground)

Constraints

ASE – Architecture Diagram[2]



ASE – CASPER

- Generates missions operations plans from goals[2].
  - Wide range of operations scenarios based on a deep model of S/C and mission constraints.
    - Represents operations (S/C and mission) constraints in a general modeling language.
      - ~700 plain text lines in ASPEN[4], includes comments.
  - NOTE: The goals are written on the ground and then uploaded.
- Must respect resources (data storage, CPU, etc)[2].
- Must be able to modify plan[2].
  - Determine next over flight opportunity, data storage available, power available, instrument prepared, calibration images acquired, etc.
  - In order for the S/C to respond autonomously to a science event, the S/C must be able to independently perform the mission planning function[2].

ASE – Science Algorithms

- **The science algorithms are custom code capable of doing anything possible given the operating environment.**
  - **Acquire a clean image of a specific ground target.**
    - **If current image has too much cloud cover then discard image and reschedule the observation.**
  - **Acquire an image of a volcano showing thermal activity.**
    - **If current image doesn't show any then discard image.**
    - **If current image does show thermal activity, then take as many images as possible.**
  - **Acquire an image of the sun showing coronal mass ejections.**
    - **If current image doesn't have any then discard.**
  - **Acquire a measurement detecting gamma ray bursts.**
    - **If detected then…..**
- **The purpose is to be able to modify the mission plan without ground intervention.**
  - **Increases the value of the science data.**
  - **Reduces meaningless data.**
    - **Reduce communications costs**
    - **Reduce mission operations costs**
  - **Allow opportunistic science.**
    - **Capture short lived science events.**
- **CASPER is a scheduler/planner. The real benefit is from being able to autonomously modify the schedule to take advantage of opportunities. CASPER will modify the schedule, but something else has to tell CASPER to modify the schedule.**

ASE – SCL

- Spacecraft Command Language (SCL) is a commercial product from Interface & Control System (ICS).
  - Integrates procedural programming with a real-time, forward-chaining, rule-based system[2].
  - Users define scripts in English-like manner[2].
  - Scripts are compiled on the ground and dynamically loaded and executed at an absolute or relative time[2].
  - Implements many constraint checks[2].

- Accepts the CASPER-derived plan as an input and expands the plan into low-level commands[2].
- Monitors the execution of the plan[2].

- SCL has flown on Clementine and Far Ultraviolet Spectroscopic Explorer (FUSE).
- URL is www.sclrules.com

CASPER - Testing

- CASPER was tested on Linux, PowerPC, and the EO-1 testbed running through many scenarios.
  - Linux and PowerPC tests used JPL developed simulator.
- Goal uploads are syntactically tested with ASPEN (ground version of CASPER) prior to upload.
- NOTE: For the EO-1 test, much of the work was done by people who would be considered experts in their respective areas, future applications may require a higher level of effort.
  - The EO-1 test facility was problematic. These had to be worked out prior to completing this effort (shouldn’t be held against a future project that has a good test facility).
  - About 2 FTEs.

CASPER – Future Plans

- No real future updates planned for CASPER.
  - I view this as a good thing (stable product).
  - There is some talk of expanding it for behavioral goals.
    - No real details available, but I assume this is like any other failure detection, isolation, and correction software.

ASE – Applicability

- **All satellites could use some level of on-board planning.**
  - **Imagers are easy to visualize.**
    - **I want an interesting picture (no clouds, some thermal activity, etc).**
  - **Distant satellites with low (or lower) data rates.**
    - **Only send back good science data.**
    - **Take advantage of opportunities.**
      - **Could be days/weeks to turn around new science timeline.**
        - **By then a volcano on some distant moon might not be erupting anymore or sandstorm on Mars won't be blowing anymore.**
  - **Applicability to lunar orbiter?**
    - **Nothing to obstruct images (no need to ever reschedule them).**
    - **No activity where many images would be useful (i.e. opportunities)**
      - **i.e. no new floods, no volcanoes with new thermal activity, no collapsing ice shelves, etc.**

CASPER - Applicability

- **CASPER is also a stand alone product.**
  - **Value as a stand alone product :**
    - **Less constraint checking on ground (done on-board).**
    - **Easy syntax for goal and model files (text format).**
      - **Doesn’t really look easier, but with good ground tools probably will be easier.**

- **CASPER is an interesting product. ASE is a really interesting solution.**
  - **CASPER is a scheduler/planner. The real benefit is from being able to autonomously modify the schedule to take advantage of opportunities. CASPER will modify the schedule, but something else has to tell CASPER to modify the schedule.**

ASE - Kudos

- Constraint checking is done on-board and doesn't have to be pre-planned.
  - The mission operations team doesn't have to build a huge sequence of absolute time tagged commands that satisfy every constraint.
- Belief that ASE (or ASE type functionality) is a move in the right direction.
  - Easier operations.
  - Opportunistic science.
- The satellite, Earth Observing-1 (EO-1), used NASA-developed software to make its own studies of an Antarctic volcano and is capable to tracking flooding events around the world before ground controllers tell it to [3].
  - The approach can cut lengthy time lags inherent to taking an observation, transmitting it to the ground for study, and later decisions by scientists to direct the satellite to take further measurements.
  - Planetary researchers agreed that smarter science software could be useful for future Mars rover or Jupiter probe missions, but the approach is very dependent on a previous knowledge of the area under study.

ASE – Implementation Details

- **Cost :**
  - CASPER is not really free.
    - Constraint model has to be defined in modeling language.
  - SCL is not free (commercial product).
    - Scripts have to be written. Could we get out of doing something we do now?
  - Science algorithms have be developed from scratch.
    - Would be specific to the instrument/sensor.
- **Memory :**
  - Memory utilization is fairly large (EO-1 numbers).
    - CASPER – 5 MB code, 32 MB data (CASPER code compresses fairly well, to less than 2 MB)
    - SCL – 100 KB code, 8 MB data
    - Science Algorithms – 100 KB code, 0 MB data (uses CASPER data space)
    - RAM Disk – 100 KB code, 8 MB data
- **CPU :**
  - CPU Utilization is fairly high (EO-1 numbers - not tuned for efficiency).
    - CASPER – 20% (100% of CPU for 5% of the time)
    - SCL – 10%
    - Science Algorithms – 0% (100% for 1-20 MINUTES when algorithms run)
- **Testing :**
  - I'm concerned about testing all this software.
    - Especially if we follow the usual routine of build testing and re-build testing every mission.
- **Even if above tasks run at a low priority, would I want the processor hosting my high priority ACS software running at 100% for 20 minutes? Then again, why not replace idle CPU time with something valuable.**
  - What about other back ground processing options? I.e. Orbit modeling?
    - Everybody can't be a back ground task.

Conclusion

- **The advantage of ASE is really on the ground side and science data side.**
  - Autonomously respond to fortuitous events.
  - Autonomously respond to un-fortuitous events.
  - Increased value of science data
    - Reduce meaningless data.
    - Take advantage of unforeseen or unpredictable events without ground intervention. This is the big advantage.
      - More valuable science data at a cheaper cost.
  - Relief mission operations team from mundane task of frequently uploading science timeline with all the constraints accounted for.
    - Would still have to load goals.
    - Goals have fairly easy syntax are are easy to load, change, update, etc.
      - Reduce mission operations costs.

- **An ASE type solution needs to be evaluated from the complete picture.**

Conclusion

- I think a tool like CASPER, or more appropriately a set of tools like ASE, has the potential to be of great value.
  - EO-1 is already showing this. Additional automation of ground processes would help.
- In order to reduce FSW costs, we need more off the shelf ready to deploy components.
  - An on-board goal oriented, constraint based scheduling/planning tool should be included.

CASPER – Future Trades

- **Complete EO-1 experiment.**
  - **Automate the ground processes?**
- **Carefully compare costs of the new EO-1 mission operations with the old EO-1 mission operations.**
- **Carefully evaluate CASPER functionality.**
  - **What is valuable?**
  - **What is marketing hype?**
- **Carefully estimate costs to flight side.**
  - **Include H/W costs as well (i.e. additional EEPROM).**
- **A goal oriented, constraint based scheduler/planning tool is only of value if you’re going to use it.**

CASPER - References

- [1] – http://planning.jpl.nasa.gov/public/planning/casper/
- [2] – R. Sherwood, S. Chien, D. Tran, B. Cichy, R. Castano, A. Davies, G. Rabideau, “Next Generation Autonomous Operations on a Current Generation Satellite”, 5th International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations (RCSGSO 2003). Pasadena, CA. July 2003
- [3] – www.space.com
- [4] - R. Sherwood, A. Govindjee, D. Yan, G. Rabideau, S. Chien, A. Fukunaga, "Using ASPEN to Automate EO-1 Activity Planning," *Proceedings of the 1998 IEEE Aerospace Conference*, Aspen, CO, March 1998.

CASPER – Points of Contact

- **Dr. Steve Chien**
  - Steve.Chien@jpl.nasa.gov
  - **818-393-5320**

- **Daniel Tran**
  - dannyt@aig.jpl.nasa.gov
  - **818-393-2677**

Acronym List

- 3CS - Three Corner Sat
- ASE - Autonomous Sciencecraft Experiment
- ASPEN - Automated Scheduling and Planning Environment
- CASPER - Continuous Activity Scheduling Planning Execution and Replanning
- CPU – Central Processing Unit
- DS-1 – Deep Space - 1
- EO-1 – Earth Observing - 1
- FDC – Failure, Detection, and Correction
- FSW – Flight Software
- FTE – Full Time Equivalent
- FUSE - Far Ultraviolet Spectroscopic Explorer
- GSFC – Goddard Space Flight Center
- ICS - Interface & Control System
- ITAR - International Traffic in Arms Regulations
- JMPL - Java-like Model Programming Language
- JPL – Jet Propulsion Laboratory
- KB - KiloByte
- L2 - Livingstone Version 2
- MB - MegaByte
- NASA – National Aeronautics and Space Administration
- RAM – Random Access Memory
- S/C - Spacecraft
- SCL - Spacecraft Command Language
- SCP – Stored Command Processor
- SWB – Software Bus
- TSM – Telemetry and Statistics Monitor
- URL - Uniform Resource Locator

Notes

Notes